

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-282095

(43) 公開日 平成7年(1995)10月27日

(51) Int.Cl.⁸

G 0 6 F 17/50

識別記号

庁内整理番号

F I

技術表示箇所

7623-5L

G 0 6 F 15/ 60

3 6 0 K

審査請求 未請求 請求項の数 3 O L (全 16 頁)

(21) 出願番号 特願平6-70383

(22) 出願日 平成6年(1994)4月8日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72) 発明者 藤澤 久典

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(74) 代理人 弁理士 長谷川 文廣 (外2名)

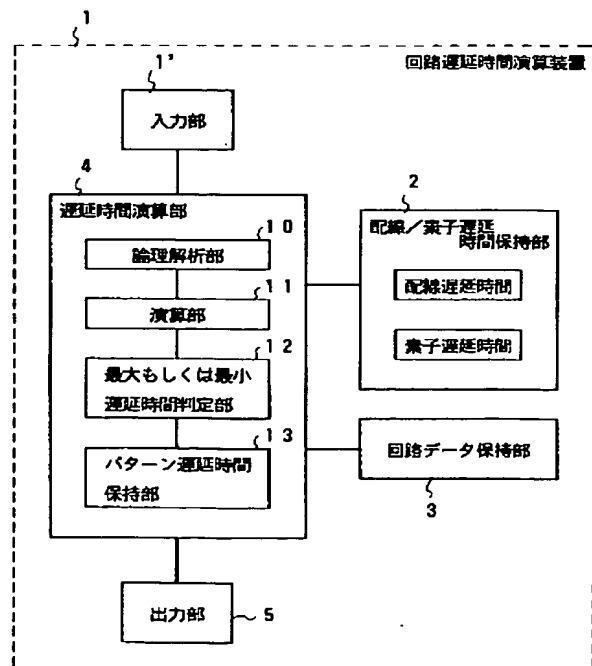
(54) 【発明の名称】 回路遅延時間演算装置

(57) 【要約】

【目的】 組合せ回路の遅延時間を演算する遅延時間演算装置に関し、高速にかつ正確に最大もしくは最小遅延時間を算出することを目的とする。

【構成】 回路データを入力する入力部(1')と、配線および素子の遅延時間を保持する配線／素子遅延時間保持部(2)と、回路データを保持する回路データ保持部(3)と、回路の遅延時間を演算する遅延時間演算部(4)と、回路の遅延時間の演算結果を出力する出力部(5)とを備え、遅延時間演算部(4)は回路に入力される外部入力パターンの論理が回路の各ノードに伝播する論理を解析する論理解析部(10)を備え、論理解析部(10)は、ノードの論理を与える外部入力パターンの有無を判定し、ノードの論理を実現する外部入力パターンが存在する論理のみにより最大遅延時間を算出する構成をもつ。

本発明の基本構成



【特許請求の範囲】

【請求項1】 回路データを入力する入力部(1')と、配線および素子の遅延時間を保持する配線／素子遅延時間保持部(2)と、回路データを保持する回路データ保持部(3)と、回路の遅延時間を演算する遅延時間演算部(4)と、回路の遅延時間の演算結果を出力する出力部(5)とを備え、該遅延時間演算部(4)は回路に入力される外部入力パターンの論理が回路の各ノードに伝播する論理を解析する論理解析部(10)を備え、該論理解析部(10)は、ノードの論理を与える外部入力パターンの有無を判定し、該ノードの論理を実現する外部入力パターンが存在する論理のみにより最大遅延時間を算出することを特徴とする回路遅延時間演算装置。

【請求項2】 請求項1において、該論理解析部(10)は、ノード毎に該ノードの論理を与える外部入力パターンの論理を求める論理式もしくは論理情報を保持し、素子の出力論理を与える素子の入力側のノードの論理を実現する外部入力パターンの有無を該素子の入力側のノードの論理式もしくは論理情報に基づいて判定することを特徴とする回路遅延時間演算装置。

【請求項3】 回路データを入力する入力部(1')と、配線および素子の遅延時間を保持する配線／素子遅延時間保持部(2)と、回路データを保持する回路データ保持部(3)と、回路の遅延時間を演算する遅延時間演算部(4)と、回路の遅延時間の演算結果を出力する出力部(5)とを備え、該遅延時間演算部(4)は、二分決定グラフ法により素子の入力端子および配線等のノードの論理を実現する外部入力パターンを求める論理解析部(10)を備え、該二分決定グラフにより各ノードのグラフ求め、求めたグラフの葉にその葉の論理を伝播する遅延時間を付加し、対象ノードと該対象ノードの前段ノードとの論理関係および前段のノードの遅延時間に基づいて対象ノードのグラフと遅延時間を求めることを特徴とする回路遅延時間演算装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、組合せ回路の遅延時間を演算する回路遅延時間演算装置に関する。コンピュータを利用して回路設計する場合、回路の入出力間の遅延時間を求める必要がある。回路が複雑な場合にはこのような遅延時間の計算は膨大な計算量を必要とするので、効率的な演算装置が望まれる。

【0002】

【従来の技術】組合せ回路の入出力の遅延時間を算出する従来の方法は、次の通りである。まず、各ゲートの入出力端子間の最大（もしくは最小）遅延時間とゲートの最大（もしくは最小）遅延時間を求めておく（遅延時間は最大遅延時間と、最小遅延時間を求める場合の2通りがあるが、以下、最大遅延時間について説明する。最小遅延時間についても方法は同じである）。

【0003】図18を参照して従来の遅延時間算出方法を説明する。図18において、110はアンドゲートである。

【0004】111はインバータである。112は前段回路である。 a_1 、 a_3 は外部入力端子である。

【0005】 a_2 、 a_4 はゲート入力端子であって、アンドゲート110の入力端子である。 a_5 はゲート出力端子であって、アンドゲート110の出力端子である。

【0006】 a_6 はゲート入力端子であって、インバータ111の入力端子である。 a_7 はゲート出力端子であって、インバータ111の出力端子である。 a_8 は外部出力端子である。

【0007】外部入力端子 a_1 、 a_3 と外部出力端子 a_8 間の遅延時間を算出する。アンドゲート110の a_2 、 a_5 間および a_4 、 a_5 間およびインバータ111の a_6 、 a_7 間の遅延時間はあらかじめ求めておく。

また、配線 a_1 a_2 、配線 a_3 a_4 、配線 a_5 a_6 、配線 a_7 a_8 の遅延時間もあらかじめ求めておく。

【0008】まず、各外部入力端子 a_1 、 a_3 から対象ゲート110、111のゲート入力端子までの最大遅延時間にそのゲート入力端子からそのゲート出力端子までの遅延時間を加算したもののうち最大のものをその対象ゲートのゲート出力端子までの最大遅延時間とする。

【0009】例えば図18の場合、 a_1 a_2 間の遅延時間に a_2 a_5 の遅延時間を加算した遅延時間と、 a_3 a_4 間の遅延時間に a_4 a_5 の遅延時間を加算した遅延時間のうち大きい方を a_5 の最大遅延時間 t_5 とする。

【0010】次に、求めたゲート出力端子 a_5 の最大遅延時間 t_5 に、そのゲート出力端子 a_5 の接続先のゲート入力端子 a_6 までの遅延時間を加算したものをそのゲート入力端子 a_6 の最大遅延時間 t_6 とする。

【0011】さらに、インバータ111に対して同様の手順で遅延時間を求め、外部入力端子 a_1 、 a_3 から外部出力端子 a_8 までの最大遅延時間を求める。

【0012】

【発明が解決しようとする課題】従来の回路の遅延時間算出方法は、ゲート出力端子の遅延時間を算出する場合、対象ゲートのゲート入力端子の論理パターン（ゲート入力パターン）を全ての論理の組み合わせに基づいて求めていた。

【0013】例えば、図18の回路の場合に、外部入力端子 a_1 、 a_3 の前に別の回路（例えばEOR）の出力は（0, 0）か（1, 1）でしかないものとする。このとき、アンドゲート110での遅延時間は、ゲート入力パターンが（0, 0）、（1, 1）である場合のみを考慮して最大遅延時間を算出すれば良いが、従来は、それ以外のあらゆる論理の組み合わせのパターンに基づいて最大遅延時間を求めていた。

【0014】そのため、有り得ないゲート入力パターンに対しても遅延時間を計算することとなり、実際以上に

最大遅延時間を過大に見積もったり、最小遅延時間を過小に見積もったりすることがあった。また、演算速度を低下させてもいた。

【0015】本発明は、高速にかつ正確に最大もしくは最小遅延時間を算出することのできる回路遅延時間演算装置を提供することを目的とする。

【0016】

【課題を解決するための手段】本発明は、各ノード（外部入力端子、ゲート入力端子、ゲート出力端子、外部出力端子、配線端子等）の論理値を実現する対象入力端子の論理パターンを記憶し、直前に接続しているノードの遅延時間からそのノードの遅延時間を計算するにあたっては、実際に実現するパターンのみを考慮して計算するようにした。

【0017】前段のノードから、対象ノードの遅延時間を計算する方法は次の通りである。ノードがゲート出力端子の場合、まずゲート入力端子の論理値の組み合わせに対して、その組み合わせが実現するような外部入力パターン（外部入力端子の入力パターン）が存在するかどうかチェックする。もし存在すればその外部入力パターンの組み合わせに対するゲート出力端子の論理および遅延時間を計算する。もし、要求している遅延時間が最大遅延時間であれば、当ゲートの論理を実現するゲート入力端子の各ゲート入力パターンの最大遅延の中から、最大遅延のものをその論理の最大遅延とする。また、要求している遅延時間が最小遅延であれば、当ゲートの論理を実現するゲート入力端子の各パターンの最小遅延のものをその論理の最小遅延とする。

【0018】一方、ノードがゲート入力端子の場合には、当ノードに繋がっているゲート出力端子の遅延に、配線遅延を加えたものを、当ノードの遅延時間とする。配線による伝播では論理は変化しないので各論理を実現する外部入力パターンの組に変化はない。

【0019】また、本発明は、このように遅延時間を求めるために必要なノード論理の判定を論理式もしくは論理情報の記憶による場合の他に、二分決定グラフ法により必要な論理のみを求める場合の両方で実現した。

【0020】図1の本発明の基本構成の説明に先立ち、二分決定グラフ法による場合について説明する。図2(a)は二分決定グラフ法の説明図である（ナンドゲートの場合）。

【0021】図2(a)に示すように、二分決定グラフ（以下、グラフと称する）は節（○）と枝（節（○）と葉（正方形）を結ぶ線）、葉（正方形）からなっている。節は入力変数に対応しており、二本の枝を持っている。二本の枝は各節の入力変数の論理値に対応しており、別の節または葉に繋がっている。入力変数の間には優先順位が付けられており、もとの節（A）の入力変数の方が枝を介して接続している節（B）の入力変数より優先順位が高いものとする。また、葉は、考慮している

ノードの論理値の最大または最小遅延情報を持っている。二分決定グラフ法はルートと呼ぶ節をもっており、ルートから葉までの経路（優先順位の低い変数（優先順位の低い節の変数）から高い変数（優先順位の高い節の変数）の節へ行くような経路は考慮しない）は、その経路の節が経路上の論理をとった場合に、到達した葉に対応した論理値を持つことを表している。枝に沿って記述してある数字は、その枝をもつ節の入力変数の論理を表している。正方形（葉）の中は出力論理で、その下の括弧で括られた数値は遅延時間を示している。入力変数の優先順位はAの方が高い。

【0022】前段のノードのグラフから、当ノードのグラフを作成する手法は以下の通りである。ノードがゲート出力端子の場合、まずゲート入力端子のグラフに対してゲートの論理演算を行うことにより、当ノードの論理値を表現するグラフを作成する（作成方法は（「Graph-Based Algorithms for Boolean Function Manipulation”, Randal E. Bryant, IEEE Transaction on Computers, Vol. C-35, No8, pp677~691, 1986, 8」参照）。グラフ同士の演算により新しいグラフを作成する過程において、新しく得られるグラフの葉の遅延時間については、その葉が得られる（その葉の論理を実現する）ゲートの入力端子の論理の組み合わせパターンが入力した際の最大（最小）遅延時間を、その葉の遅延時間とする。また、葉のリダクション（同じ論理をもつ葉を一つにまとめる（図2(b)参照）場合には、もし要求している遅延時間が最大遅延時間であれば、最大遅延のものをその論理の最大遅延とし、要求している遅延が最小遅延であれば、最小遅延のものをその論理の最小遅延とする。この処理をグラフ演算に付加することにより該当ノードのグラフと遅延時間が求められる。

【0023】一方ノードがゲートの入力端子の場合には、当ノードに繋がっているゲートの出力端子のグラフの各論理の遅延に配線による遅延を加えたものを、当ノードのグラフとする。

【0024】図1は本発明の基本構成を示す図である。図1において、1は回路遅延時間演算装置である。

【0025】2は配線／素子遅延時間保持部であって、配線の遅延時間と素子（ゲート等）の遅延時間を保持するものである。3は回路データ保持部であって、ノードの結合情報であるネットリスト、回路特性を示すパラメータ等の回路データを保持するものである。

【0026】4は遅延時間演算部であって、回路の遅延時間を演算するものである。5は出力部であって、演算結果を出力するものである。遅延時間演算部4において、10は論理解析部であって、各ノードにおける論理を解析するものであり、各ノードの論理を与える外部入力パターン（外部入力端子に与えられる入力変数の組）

の論理を求めるための論理式もしくは論理情報(真理値表等)をノード毎に求め、外部入力端子の論理を実現する外部入力パターンの有無を判定するものである。あるいは、二分決定グラフを求めるものである。

【0027】11は演算部であって、ノードの有効パターン(ノードの論理を実現する外部入力パターンが存在するノードの論理パターン)に基づいて遅延時間を算出するものである。

【0028】12は最大もしくは最小遅延時間判定部であって、計算された遅延時間最大もしくは最小遅延時間を求めるものである。13はパターン遅延時間保持部であって、ノードの論理を与える外部入力パターンを求めるための論理式もしくは論理情報を保持するとともに、最大遅延時間を保持するものである。

【0029】

【作用】図1の本発明の基本構成の動作を説明する。図1において、例えば、外部出力端子から外部入力端子に向かって各ノードを遡り、論理解析部10は外部入力端子から近いノードから順番に対象ノードの遅延時間および論理式もしくは論理情報(ノードの論理を実現する外部入力パターンを求めるための論理式もしくは論理情報であって、以後単に論理式もしくは論理情報と称する)を求める。そして、ノードの論理特性により決められるノードの論理を実現する外部入力パターンの有無を判定する。そして、演算部11は、ノードの遅延時間を求めるのに必要なノードの論理のうち外部入力パターンの存在するノードの論理についてのみ、その論理が伝播する遅延時間を算出する。最大もしくは最小遅延時間判定部12は求められた遅延時間のうち対象ノードで最大(もしくは最小)のものを求める。パターン遅延時間保持部13はその対象ノードの論理式もしくは論理情報および最大遅延時間を保持する。

【0030】以上の処理を外部入力端子に近いノードから外部出力端子まで順番に繰り返し、外部出力端子の最大もしくは最小遅延時間を求め、出力部5に出力する。図3の回路により具体的に説明する。ゲートG₄に着目する。ゲートG₄のゲート入力端子a₁₁, a₁₃(いずれもノード)について考える。外部入力パターン(a, b, c)に対してa₁₁の論理を1とする論理式は $(-a) \cdot (-c)$ である(-は否定を表す。例えば $(-b)$ はbの否定論理である)。即ち、 $(-a) \cdot (-c) = 1$ とするa, cの論理はa=0かつc=0である。従って、論理式 $(-a) \cdot (-c)$ によりa₁₁の論理1もしくは論理0を与える外部入力パターン(a, b, c)の組を求めることができる。同様にa₁₃の論理式は $(-b) \cdot c$ である。

【0031】一方、ゲートG₄の出力を1とする入力端子a₁₁, a₁₃のとる論理パターンは(0, 0)と(1, 1)である。そして、(a₁₁, a₁₃)を(0, 0)とする外部入力パターン(a, b, c)は、次の論理式

$(-((-a) \cdot (-c))) \cdot (-((-b) \cdot c)) = a \cdot b + a \cdot (-c) + b \cdot c = 1$ とするものである。

【0032】論理解析部10は、(a, b, c)の組み合わせパターンから、そのような有効な外部入力パターンは(1, *, 0)もしくは(*, 1, 1)であることを判定する(*は0, 1いずれでも良い)である。そして、G₄の出力を1とする場合の遅延時間の演算はこの二通りのパターンについてのみなされ、ゲートG₄の遅延時間が求められる。また、ゲートG₄の入力(a₁₁, a₁₃)を(1, 1)とする論理式は、

$$((-a) \cdot (-c)) \cdot ((-b) \cdot c) = 1$$

である。この式は(a, b, c)の全てのパターンに対して恒等的に0であるので、(a₁₁, a₁₃)を(1, 1)とする外部入力パターンは存在しない。従って、論理解析部10は上記の論理式を満たす外部入力パターンは存在しないことを判定し、遅延時間判定部12はそのようなゲート入力パターンについては遅延時間の計算は行わない(なお、図3の回路の遅延時間の算出方法の詳細については後述する)。さらに、G₄の出力を0とする場合についても同様に外部入力パターンの有無を判定し、有効なゲート入力パターンについてのみ遅延時間を求める。

【0033】本発明によれば、素子の入力論理パターンについて、実際に発生する有効なゲート入力パターンについてのみ遅延時間を求めるので、少ない演算回数で正確な遅延時間を算出することが可能となる。

【0034】

【実施例】本発明の実施例の装置構成を説明する前に、図3を参照して本発明の回路遅延時間演算方法について説明する。

【0035】図3は本発明の実施例を説明するための回路例である。図3において、G₁はゲートであって、インバータである。

【0036】G₂はゲートであって、ノアゲートである。G₃はゲートであって、ノアゲートである。G₄はゲートであって、排他的論理回路である。

【0037】A(=a₁)は外部入力端子(ノード)である。B(=a₆)は外部入力端子(ノード)である。C(=a₃)は外部端子(ノード)Aの入力である。

【0038】a₂, a₄はゲートG₂のゲート入力端子(ノード)である。a₁₀はゲートG₂のゲート出力端子(ノード)である。a₅はゲートG₁のゲート入力端子(ノード)である。

【0039】a₈はゲートG₁のゲート出力端子(ノード)である。a₇, a₉はゲートG₃のゲート入力端子(ノード)である。a₁₂はゲートG₃のゲート出力端子(ノード)である。

【0040】a₁₁, a₁₃はゲートG₄のゲート入力端子(ノード)である。a₁₄はゲートG₄のゲート出力端子

(ノード)である。OUT (= a_{15}) は外部出力端子である。

【0041】説明を簡単にするために、遅延時間の計算において、各ゲート間の配線時間は1とする。また、入力に変化したときに出力がLからHに変化する遅延をアップ遅延と呼ぶ。入力に変化したとき、出力がHからLに変化する遅延をダウン遅延と呼ぶ。また、インバクタ G_1 のアップ遅延は8、ダウン遅延は6とする。ノアゲート G_2 、 G_3 のアップ遅延は10、ダウン遅延は7とする。排他的論理回路 G_4 のアップは9、ダウンは7であるとする。

【0042】以下ノード a_j のアップ遅延を $a_j \uparrow$ 、ダウンを $a_j \downarrow$ で表すものとする。本発明の装置構成を説明する前に、論理式により有効なゲート入力パターン(有効なノードの論理)を求める場合の遅延時間演算法について説明する(二分決定グラフによらない場合)。

【0043】外部入力端子A、B、Cから外部出力端子(OUT)までの最大遅延時間を計算する。 a_{15} から入力端子側にパスを辿る。 a_{11} と a_{13} はともに、未処理なので、まず a_{13} を選択する。再び未処理の入力をもつノアゲート G_3 にたどりつく。そこで、 a_7 を選択して、さらに溯り、外部入力ノード a_6 に到達する。その遅延時間は0である。 a_7 の遅延時間は a_6 の遅延時間に $a_6 - a_7$ 間の配線遅延を加えたものである。

【0044】 $a_7 \uparrow = a_6 \uparrow + 1 = 1$

$a_7 \downarrow = a_6 \downarrow + 1 = 1$

である。

【0045】また、ノード a_7 の論理式は、

b

である。b=1のとき $a_7 = 1$ であり、b=0のとき $a_7 = 0$ である。 a_7 のこの論理式もしくは論理条件(真理値表等)を a_7 に対応付けて保持する。

【0046】次は、 a_9 の遅延時間を計算する。 a_9 から溯ると、外部入力ノード a_3 に到達する。ノード a_3 の遅延時間は0であるから、 a_5 のノードの遅延時間は、 a_3 の遅延時間に配線遅延1を加えた

$a_5 \uparrow = a_3 \uparrow + 1 = 1$

$a_5 \downarrow = a_3 \downarrow + 1 = 1$

a_5 の論理式は

c

である。ノード a_5 にこの論理式もしくは論理情報を保持する。

【0047】さらに、出力側に戻って a_8 の遅延時間を計算する。 a_8 はインバクタの出力であるから、 a_8 が1となるのは入力ノード a_5 が0の時である。この時、入力パターンはc=0の時であるから、遅延時間は

$a_8 \uparrow = a_5 \downarrow + 8 = 9$

である。また、 a_8 が0となるのは入力ノード a_5 が1の時である。この時の入力パターンはc=1の時である

から、遅延時間は、

$a_8 \downarrow = a_5 \uparrow + 6 = 7$

となる。また、 a_8 が1となる時の入力パターンはc=0であるから、論理式は

(-c)

である。ノード a_8 にこの論理式もしくは論理情報を保持する。

【0048】そこで、配線遅延1をアップ遅延とダウン遅延に加えると、

10 $a_9 \uparrow = a_8 \uparrow + 1 = 10$

$a_9 \downarrow = a_8 \downarrow + 1 = 8$

である。

【0049】 a_7 、 a_9 の遅延時間が得られたので、 a_{12} の遅延時間を計算する。ゲート入力端子 (a_7 、 a_9) の組合せは(0, 0)、(1, 0)、(1, 1)、(0, 1)である。 a_9 の論理式は(-c)、 a_7 の論理式はbである。従って、 $a_{12} = 1$ とする G_3 の入力側の論理条件は、 $-(b + (-c)) = 1$ である。この式から $a_{12} = 1$ とする外部入力パターン(*, 0, 1)が求まる。従って $a_{12} \uparrow$ は $a_7 \downarrow$ と $a_9 \downarrow$ に各々遅延を加えたものの最大のものとなる。また G_3 の出力側を0とする入力側の論理条件は、 $-(b + (-c)) = 0$ である。その外部入力パターンは(*, 1, *)または(*, *, 0)である。従って $a_{12} \uparrow$ はゲート入力のパターン(1, *)と(*, 1)の遅延の遅いものとなる。

【0050】従って、 $a_{12} \uparrow$ 、 $a_{12} \downarrow$ は次のように求まる。

$a_{12} \uparrow = \text{MAX}(a_7 \downarrow + 10, a_9 \downarrow + 10) = 18$

30 $a_{12} \downarrow = \text{MAX}(a_7 \uparrow + 7, a_9 \uparrow + 7) = 17$

である(但し、MAXは括弧内の値のうち大きい方を選択することを表す)。

【0051】 a_{13} は a_{12} に $a_{12} - a_{13}$ 間の配線遅延時間を加算したものである。即ち、

$a_{13} \uparrow = a_{12} \uparrow + 1$

$a_{13} \downarrow = a_{12} \downarrow + 1$

である。

【0052】次に a_{14} の遅延時間を計算する。 a_{14} の論理値が1となる場合について考える。このような出力となる(a_{11} 、 a_{13})の組は(0, 0)と(1, 1)である。

【0053】 a_{11} 、と a_{13} の論理式はそれぞれ $(-a) \cdot (-c)$ 、 $(-b) \cdot c$ と表すことができる。そこで、(0, 0)のパターンを実現する論理式は

$(-((-a) \cdot (-c))) \cdot (-((-b) \cdot c))$

$= (a + c) \cdot (b + (-c))$

$= (a \cdot b) + a \cdot (-c) + b \cdot c = a \cdot (-c) + b \cdot c$

であるから

$$a \cdot (-c) + b \cdot c = 1$$

とするものである(前述)。

【0054】この関係から、 G_4 の入力が(0, 0)となる組合せは $(a, b, c) = (1, *, 0), (*, 1, 1)$ が求まる(*は0でも1でも良いことを表している)。

【0055】一方、 G_4 の入力が(1, 1)となる論理式は

$$(-a) \cdot (-c) \cdot (-b) \cdot c = 1$$

である。しかし、 $(-a) \cdot (-c) \cdot (-b) \cdot c$ はどのような a, b, c の組合せに対しても0であるので、 $(-a) \cdot (-c) \cdot (-b) \cdot c = 1$ となる組合せはなく、 G_4 の入力を(1, 1)とする入力 (a, b, c) は存在しない。

【0056】従って、 $a_{11} \uparrow, a_{13} \uparrow$ は考慮する必要がない。 a_{14} のダウン遅延は、

$$a_{14} \downarrow = \text{MAX}(a_{11} \downarrow + 7, a_{13} \downarrow + 7) = 25$$

と求めるだけで良い。

【0057】仮に、従来のように入力パターンの組合せを考慮しないと、

$$a_{14} \downarrow = \text{MAX}(\text{MAX}(a_{11} \downarrow + 7, a_{13} \downarrow + 7), \text{MAX}(a_{11} \uparrow + 7, a_{13} \uparrow + 7)) = 26$$

となる。

【0058】 $a_{14} = 0$ となる場合も、同様に、このような出力値を与える G_4 の入力値は(0, 1),

(1, 0)であり、このようなパターンを与える入力パターンの組合せは(0, *, 0), (*, 0, 1)である。前者のパターンは $a_{11} = 0, a_{13} = 1$ となり、後者のパターンは $a_{11} = 1, a_{13} = 0$ となる。従って、アップ遅延時間は、次のように求まる。

【0059】 $a_{14} \downarrow = \text{MAX}(\text{MAX}(a_{11} \downarrow + 9, a_{13} \uparrow + 9), \text{MAX}(a_{11} \uparrow + 9, a_{13} \downarrow + 9)) = 28$ である。 a_{15} は a_{14} に配線遅延時間を加えたものであり、出力ノードにおけるアップ遅延は29, ダウン遅延は26が得られる。

【0060】図4は本発明の実施例構成1を示す。図4は論理式もしくは論理情報の記憶により有効なゲート入力パターンを求める場合の構成を示す。図4において、20は制御部であって、各ブロック間のデータの流れを制御するものである。

【0061】21は入力部であって、回路データを読み込んで回路データ記憶部23に書き込むものである。22は遅延時間演算手段である。

【0062】23は回路データ記憶部であって、ゲートパラメータ、配線パラメータ、ネットリスト等を保持するものである(図1の回路データ保持手段に相当する)。24は配線/ゲート遅延時間記憶部であって、配線遅延時間、ゲート遅延時間等を保持するものである(図1の配線/素子遅延時間保持部に相当する)。

【0063】25はパターン/遅延時間記憶部であって

て、有効な外部入力パターン、ノードの論理式(もしくは真理値表等の論理情報)、ノードの遅延時間等を保持するものである(図1のパターン/遅延時間保持部に相当する)。

【0064】30は配線/ゲート遅延時間計算部であって、回路データ記憶部23に記憶されている回路データ(配線長、配線幅、素子の特性、ゲート幅等)に基づいて、配線遅延時間、ゲート遅延時間を計算するものである。

【0065】31はパス解析部であって、回路データに基づいて回路のパスを出力側から入力側にたどり、回路解析を行うものである。32は遅延時間演算部である。

【0066】33は論理解析部であって、論理パターンの解析を行い、有効なゲート入力パターン、外部入力パターンを求めるものである。34は演算部であって、遅延時間を計算するものである。

【0067】35は最大遅延時間判定部であって、最大遅延時間を求めるものである。36は出力部である。図5は本発明の実施例構成1のフローを示す。

20 【0068】S1 配線/ゲート遅延時間計算部30は配線、ゲートの遅延時間を求め、配線/ゲート遅延時間記憶部24に格納する。

S2 パス解析部31は遅延時間の演算対象とするノード(対象ノード)を求める。

【0069】S3 処理Aを実行する(処理Aは図6参照)。図6は処理Aのフローである。

S1 遅延時間演算部32において、ノードは外部入力端子、ゲート端子、ゲート入力端子のどれであるかを判定する。外部入力端子(図3の a_1, a_3, a_6)であればS2に進む。ゲート出力端子(図3の $a_8, a_{10}, a_{12}, a_{14}$)であればS3に進む。ゲート入力端子($a_2, a_4, a_5, a_7, a_9, s_{11}, a_{13}$)であれば、S6に進む。

【0070】S2 外部入力端子であるので、ノードの遅延時間は0である。

S3 すべてのゲート入力端子の遅延時間は求められているか判定する。求められていればS5に進み、求められていなければS4に進む。

【0071】S4 遅延時間未定の入力端子ノードに対して処理Aを実行する(S6以降の処理)。

S5 ゲート遅延時間のデータからゲート出力端子の遅延時間を計算する(遅延時間の算出の詳細フローは図7で説明する)。

【0072】S6 前段の出力端子の遅延時間は求められているか判定する。求められていればS8に進み、求められていなければS7に進む。

S7 出力端子ノードに対して処理Aを実行する(S3以降の処理)。

【0073】S8 出力端子の遅延時間に対して配線遅延時間を加算したものをノードの遅延時間とする。図7

11

は本発明のゲート出力端子の遅延時間の計算のフローである(図6のS5の処理)。

【0074】S1 ゲート入力端子のすべてのパターン(ゲート入力パターン)について以下の処理を行う。

S2 ゲート入力端子の論理値を表す論理集合を求める(論理解析部の処理)。

【0075】S3 すべてのゲート入力端子の論理集合についてANDをとる(論理解析部の処理)。

S4 得られた論理集合が空集合か判定する(論理解析部の処理)。

【0076】S5 遅延時間を計算する(演算部の処理)。

S6 ゲート入力パターンによる出力端子の論理値を求める。出力端子の論理値がH(アップ遅延)であればS7に進む。出力端子の論理値がL(ダウン遅延)であればS9に進む。

【0077】S7 出力端子の論理値がHとなるパターンのこれまで得られた最大(もしくは最小)遅延と比較する(最大遅延時間判定部の処理)。最大でなければS1以降の処理を繰り返す。最大であればS8に進む。

【0078】S8 新しい遅延時間を最大(もしくは最小)遅延時間とし、S1に戻る。

S9 出力端子の論理値がLとなるパターンのこれまで得られた最大(最小)遅延と比較する。最大(もしくは最小)遅延でなければS1に戻る。最大(もしくは最小)遅延であればS10に進む。

【0079】S10 新しい遅延時間を最大遅延時間とし、S1に戻る。S1において、ゲートの全てについて処理がなされたら、処理を終了する。図8、図9は二分決定グラフ法により最大遅延時間を計算する方法を示す。

【0080】図8、図9により二分決定グラフ法により最大遅延時間を計算する方法について説明する。 a_{15} から入力側にパスを溯る。 a_{11} と a_{13} はともに、未処理なので、まず a_{13} を選択する。そして、再び未入力の入力をもつ二入力ゲート G_3 にたどりつくので、先に a_7 を選択してさらに溯り、Bに到達する。B($=a_6$)は入力ノードであり、そのグラフは図8(e)のように表される。遅延時間は0である。次に a_7 のグラフを作成する。 a_7 のグラフは a_6 のグラフに $a_6 - a_7$ 間の配線遅延を加えたものである。そこで、図8(e)のグラフの葉の各遅延時間に1を加え、図8(e)のグラフを得る。 a_7 のグラフが得られたので、次は a_9 のグラフを作成する。 a_9 から遡ると、外部入力ノード a_3 に達する。ノード a_3 のグラフは図8(c)のように表される。さらに、出力側に戻って、 a_8 のグラフを作る。 a_8 はインバータの出力であるから、入力ノードの a_5 のグラフの理論値を入れ換えたものが、 a_8 のグラフとなる。また、遅延はそれぞれの遅延にインバータの遅延を加算したものである(図8(g))。

12

【0081】 $a_8 \uparrow = a_5 \downarrow + 8 = 9$

$a_8 \downarrow = a_5 \uparrow + 6 = 7$

を求め、図8(e)のグラフを得る。

【0082】さらに、配線遅延1をアップ遅延、ダウン遅延に加えると a_9 のグラフが得られる(図9

(h))。こうして、 a_7 と a_9 のグラフが得られたので、ゲート G_3 の入出力論理を考慮して a_{12} のグラフを作成する。作成されるグラフは図9の(k)のようになる。ここで遅延は、

10 $a_{12} \uparrow = \text{MAX}(a_7 \downarrow + 10, a_9 \downarrow + 10) = 18$

$a_{12} \downarrow = \text{MAX}(a_7 \uparrow + 7, a_9 \uparrow + 7) = 17$

で計算される。 a_{13} のグラフは a_{12} のグラフに配線遅延を加えたものである。そこで、図9(l)のグラフを得る。 a_{13} が得られたので、次に a_{11} のグラフを求める。 a_{11} も a_{13} 同様にして計算すると、図9(j)のグラフが得られる。 a_{14} は a_{12} と a_{14} のグラフに配線遅延を加えたものである。そして、図9(n)のグラフが得られる。遅延時間は、

$a_{14} \downarrow = \text{MAX}(a_{11} \downarrow + 7, a_{13} \downarrow + 7) = 25$

20 $a_{14} \uparrow = \text{MAX}(\text{MAX}(a_{11} \downarrow + 9, a_{13} \uparrow + 9), \text{MAX}(a_{11} \uparrow + 9, a_{13} \downarrow + 9)) = 28$

である。

【0083】 a_{15} は a_{14} のグラフに配線遅延を加えたものである。そして、図9(n)のグラフが得られる。こうして、出力ノード a_{15} におけるアップ遅延は29、ダウン遅延は26が得られる。

【0084】図10、図11、図12は本発明の二分決定グラフ法による場合のアルゴリズムである。図10(a)はノードのデータ構造の定義である。

30 【0085】vertexはノードであって、highはHの論理値を持つ枝につながるノードを定義し、lowはLの論理値を持つ枝につながるノードを定義する。delay(遅延時間)は実数であることを定義する。

【0086】図10(b)以降、図11、図12は、二入力ゲートにおけるグラフ作成アルゴリズム(最大遅延の場合)である。図11において、51は最大遅延を求める処理である(例えば、図14の演算を行うものである(後述))。

【0087】52はグラフを求める処理である。53はメインルーチンである。メインルーチン53において、処理60により図10の処理62以降の処理を行う。そして、その処理が全てなされると処理61により図12の処理63以降のリダクションの処理を行う。

【0088】図12において、54はリダクションの処理である。55は最大遅延を選択する処理であって、葉をリダクションするときにそれぞれの保持する最大リダクションのうち最大のものを選択する処理を行う。

【0089】図13は本発明のグラフ法による場合のアルゴリズムにおける<op>の意味の説明図である。<op>はゲートの演算を表し、ANDゲートの場合は、

$A < op > B$ は図示の論理を演算する。Xは1もしくは0のいずれでも良いことを表す。

【0090】図14は本発明のグラフ法による場合のアルゴリズムにおける $< delay op >$ の意味の説明図である。 $< delay op >$ はゲート出力の遅延時間を計算する演算子を表す。図はANDゲートの場合の演算論理を示す。

【0091】 $A \cdot value$, $B \cdot value$ ともに、1の場合は出力は1であるので、A側の $delay$ とB側の $delay$ の大きい方を遅延とする。A側の $delay$ はA端子のゲートにおける $delay$ とその時の出力側のアップ遅延 $\alpha_A \uparrow$ の和であり、B側の $delay$ はB端子のゲートにおける $delay$ とその時の出力側のアップ遅延 $\alpha_B \uparrow$ の和であるので、その大きい方を選択する。

【0092】以下同様に、他の入力論理の場合は図示の論理の演算を行う。図15は本発明のグラフ法のアルゴリズムのフローチャートである。図15(a)はメインルーチンの処理である(図11の53参照)。

【0093】S1 グラフG1とG2に対して処理1($apply-step$)を適用する。

S2 得られたグラフに対して処理2を行う($reduce$)。

【0094】図15(b)は最大遅延を求める処理であって、処理1($apply-step$)である。

S1 新しいノードを作成する。

【0095】S2 グラフG1とグラフG2のルートノード($v1$ および $v2$)の値($value$)に対して演算 $< op >$ を行い、新規ノードの値とする。

S3 値がX(図13, 図14のXと異なる)でなければ、S5に進み、XであればS4に進む。

【0096】S4 グラフの作成処理をする(図16参照)。

S5 新規ノードのインデックス($index$)に $n+1$ を代入する。

S6 グラフG1とグラフG2のルートノード($v1$ および $v2$)に対して演算 $< delay op >$ を行って、遅延時間を計算する。

【0097】図16は本発明のグラフ法における場合のアルゴリズムのフローチャートであって、図15のS4の処理である。

S1 $v1$ と $v2$ を比較する。 $v1$ の $index < v2$ の $index$ であればS2に進み、 $v1$ の $index = v2$ の $index$ であればS3に進み、 $v1$ の $index > v2$ の $index$ であればS4に進む。

【0098】S5 $v1$ のインデックス($index$)を新規ノードのインデックス($index$)とする。

S6 $v2$ のインデックスを新規ノードのインデックス($index$)とする。

【0099】S7 $vlow1$ と $vhigh1$ をルート

とするグラフ処理1($apply-step$)を適用し、できたグラフを新規ノードの右枝(low)におさめる。

S8 $vlow2$ と $vhigh2$ をルートとするグラフ処理1($apply-step$)を適用し、できたグラフを新規ノードの左枝($high$)におさめる。

【0100】図17は本発明のグラフ法のアルゴリズムにおけるリダクションの処理のフローチャートである。

S1 左右の枝(low と $high$)が同一ノードを指しているノードを削除する。

【0101】S2 グラフの中のノードのうち左右の枝が互いに同じノードを指しているノードの組を検索し、存在しなければ、終了する。

S3 存在すれば、これらのノードをひとつにまとめる。

【0102】S4 これらのノードが葉($terminal node$)ならば、これらのノードの遅延時間のうち最大(最小)のものを新しいノードの遅延時間とする。

【0103】

【発明の効果】本発明によれば、実際には生じない無駄なゲート入力のパターンは計算しないので回路の最大遅延時間もしくは最小遅延時間を正確に計算できるとともに、処理が高速化される。また、二分決定グラフ法を利用した場合には、各ノードの入力パターンの解析を高速に行うことができ、正確な遅延時間を高速に算出する。

【図面の簡単な説明】

【図1】本発明の基本構成を示す図である。

【図2】二分決定グラフ法の説明図である。

【図3】回路例を示す図である。

【図4】本発明の実施例構成1を示す図である。

【図5】本発明の実施例構成1のフローを示す図である。

【図6】本発明の処理Aのフローを示す図である。

【図7】本発明の出力端子の遅延時間の計算のフローを示す図である。

【図8】二分決定グラフ法により最大遅延時間を計算する方法(実施例2)を示す図である。

【図9】二分決定グラフ法により最大遅延時間を計算する方法(実施例2)を示す図である。

【図10】本発明のグラフ法による場合のアルゴリズムを示す図である。

【図11】本発明のグラフ法による場合のアルゴリズムを示す図である。

【図12】本発明のグラフ法による場合のアルゴリズムを示す図である。

【図13】本発明のグラフ法による場合のアルゴリズムの説明図である。

【図14】本発明のグラフ法による場合のアルゴリズムの説明図である。

【図15】本発明のグラフ法のアルゴリズムのフローチャートを示す図である。

【図16】本発明のグラフ法による場合のアルゴリズムのフローチャートを示す図である。

【図17】本発明のグラフ法による場合のアルゴリズムのフローチャートを示す図である。

【図18】従来の遅延時間算出方法の説明図である。

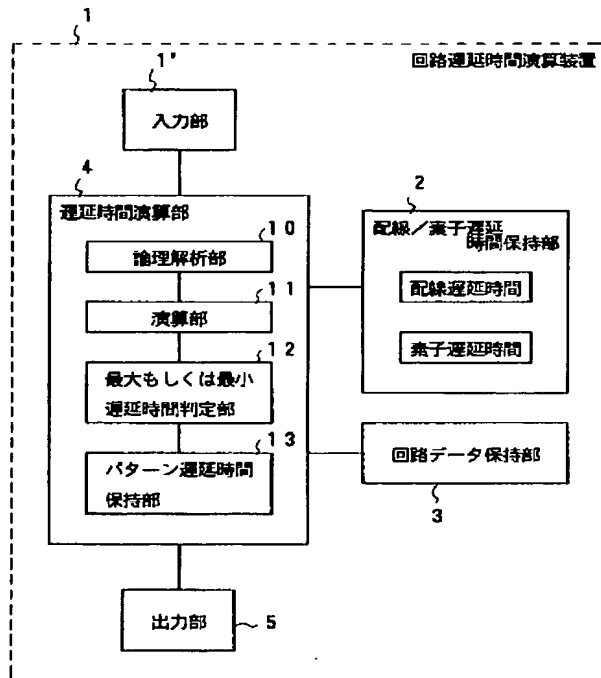
【0122】

【符号の説明】

1：回路遅延時間演算装置

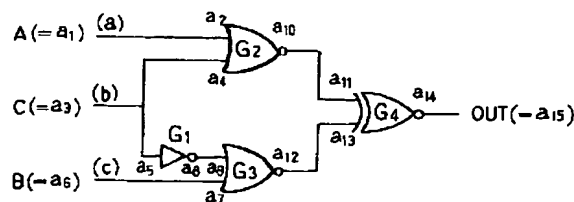
【図1】

本発明の基本構成



【図3】

回路例



1'：入力部

2：配線／素子遅延時間保持部

3：回路データ保持部

4：遅延時間演算部

5：出力部

10：論理解析部

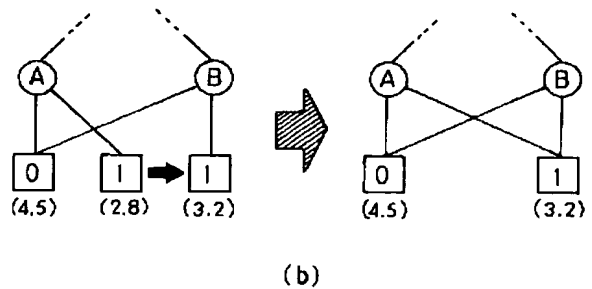
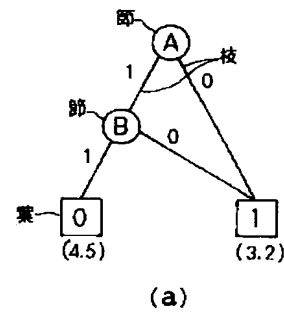
11：演算部

12：最大もしくは最小遅延時間判定部

13：パターン遅延時間保持部

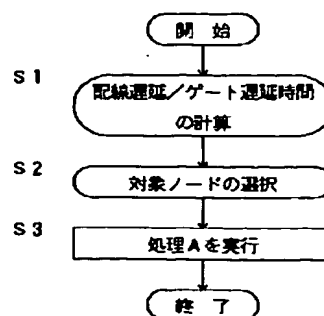
【図2】

二分決定グラフ法の説明図



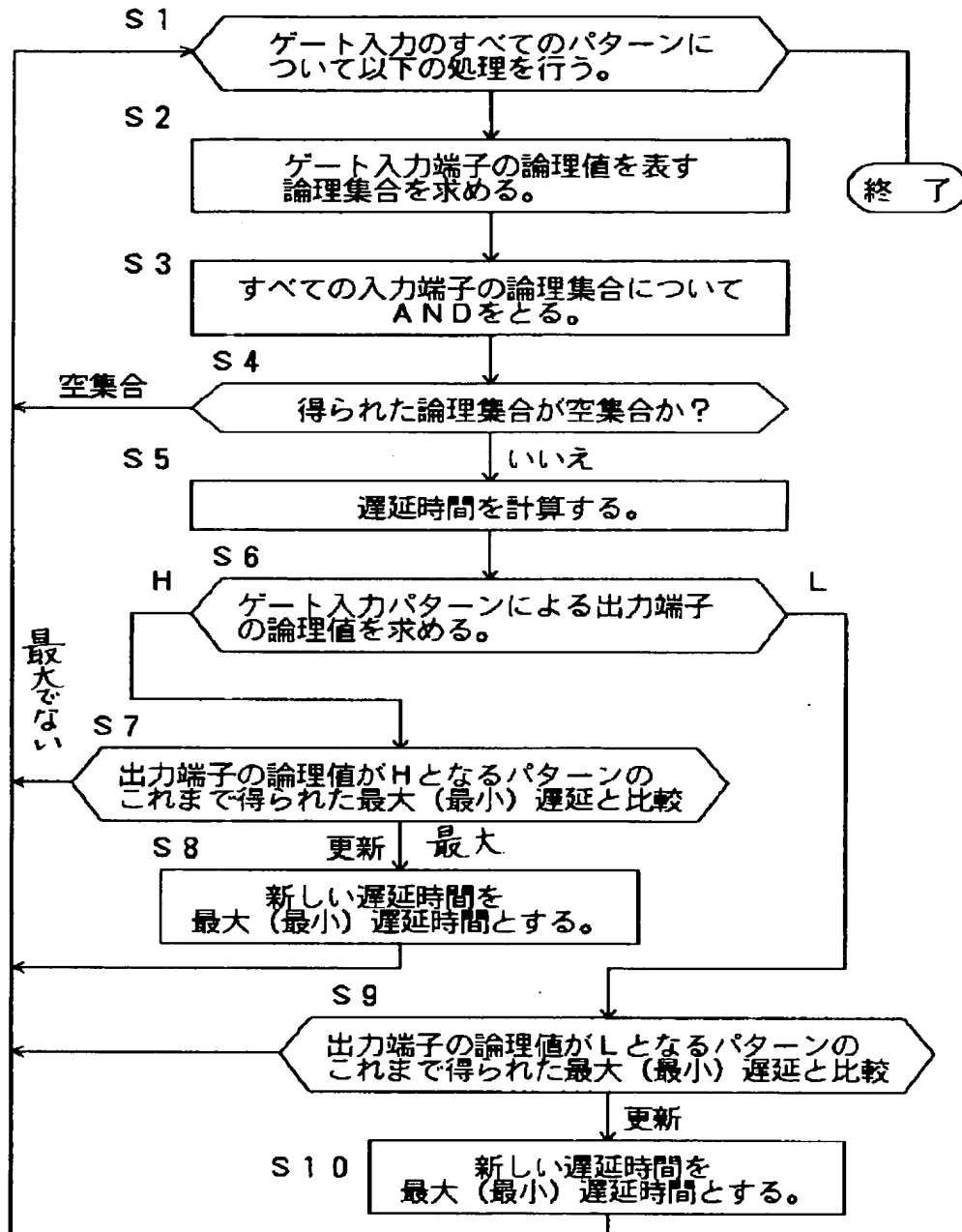
【図5】

本発明の実施例構成1のフロー



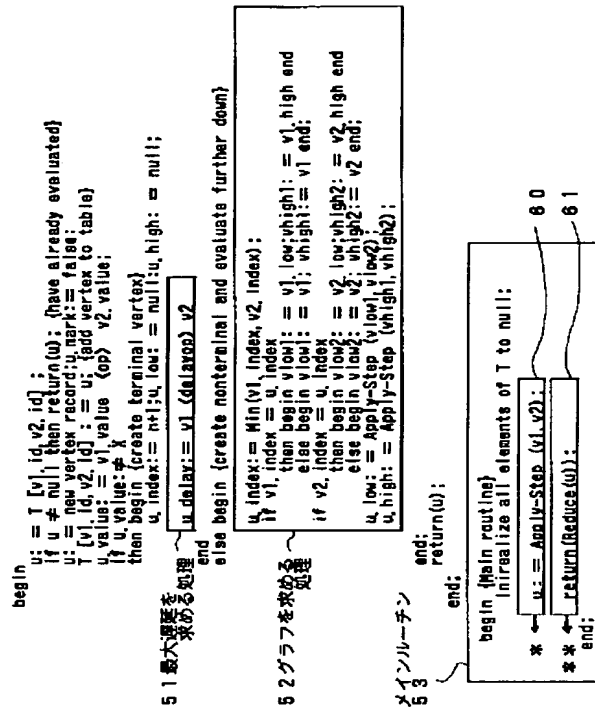
【図 7】

出力端子の遅延時間の計算のフロー



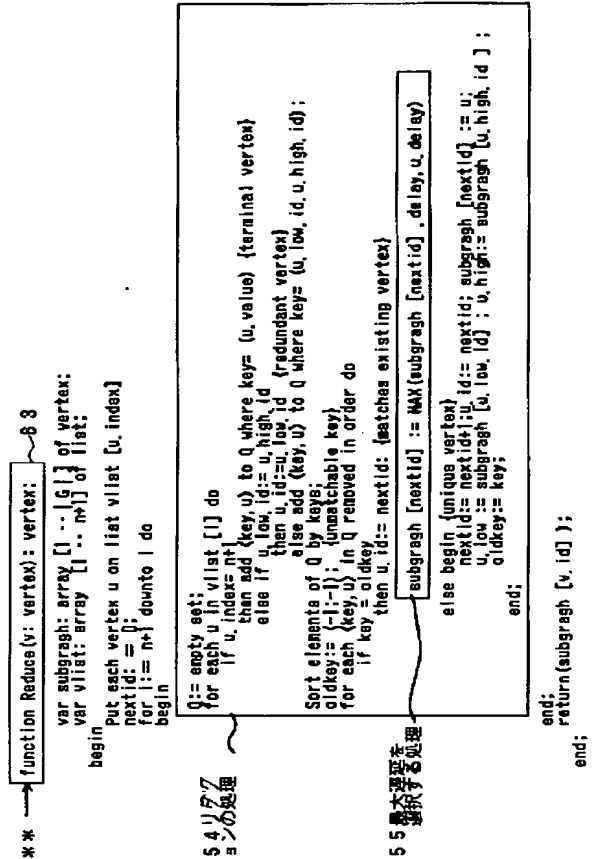
【図11】

本発明のグラフ法による場合のアルゴリズム



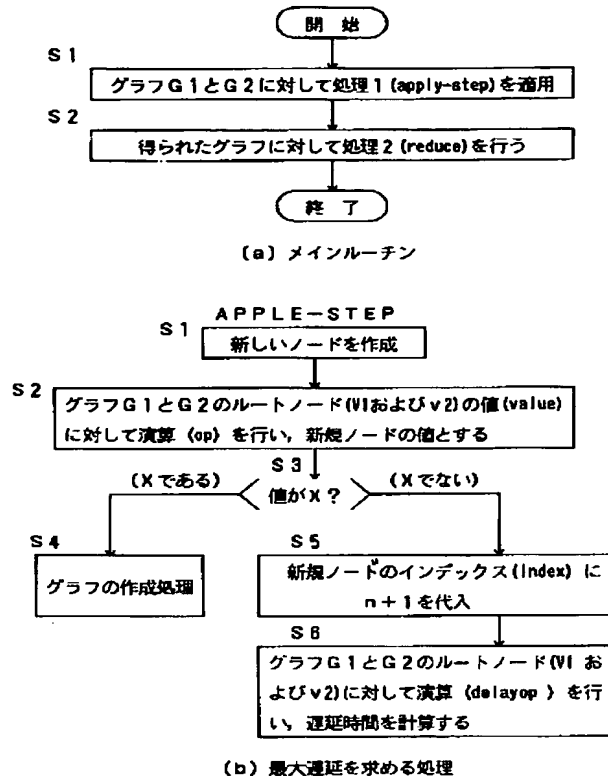
【図12】

本発明のグラフ法による場合のアルゴリズム



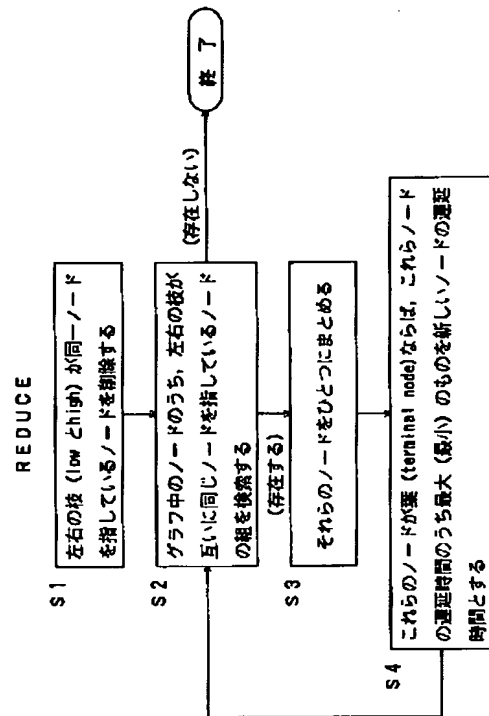
【図 15】

本発明のグラフ法のアルゴリズムのフローチャート



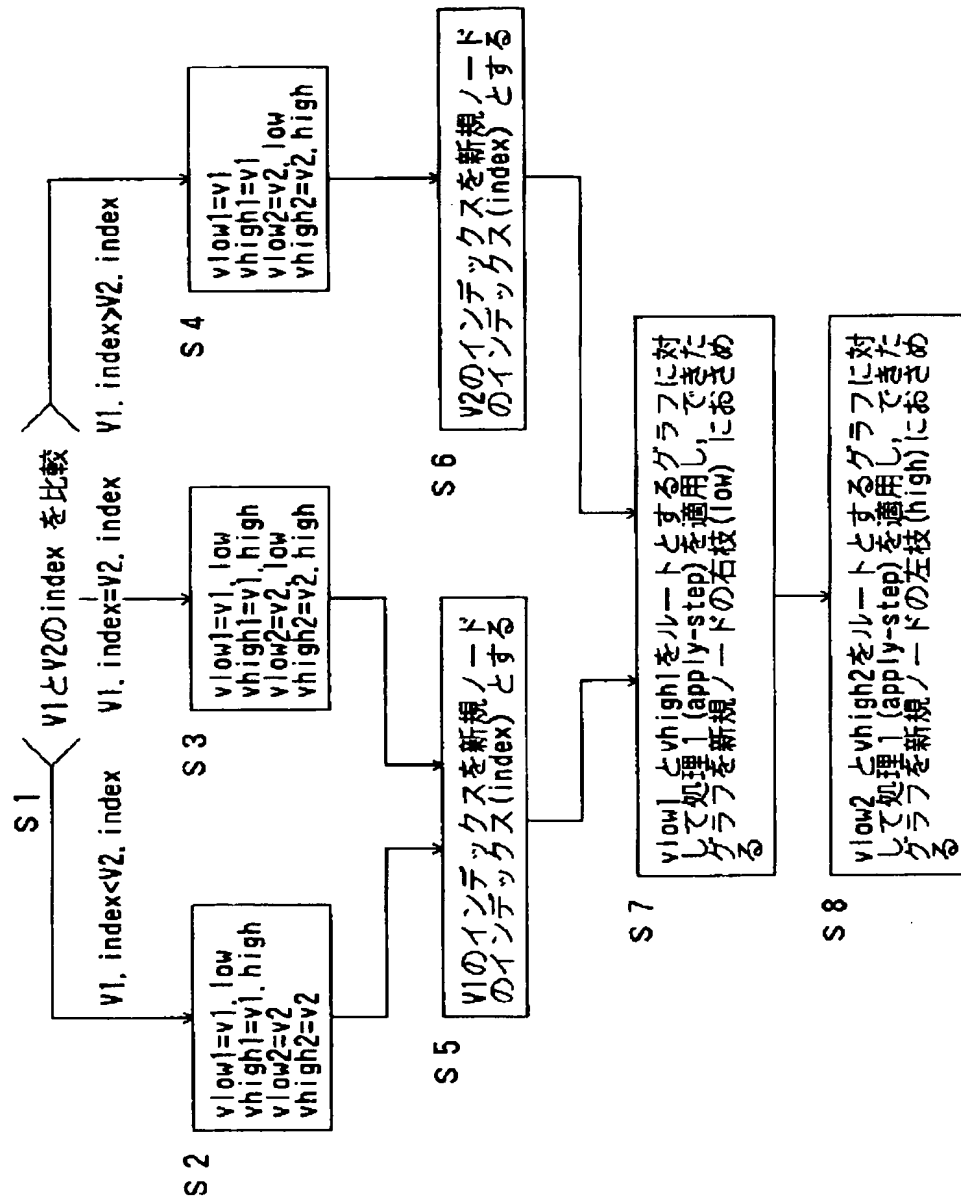
【図 17】

本発明のグラフ法による場合のアルゴリズムのフローチャート



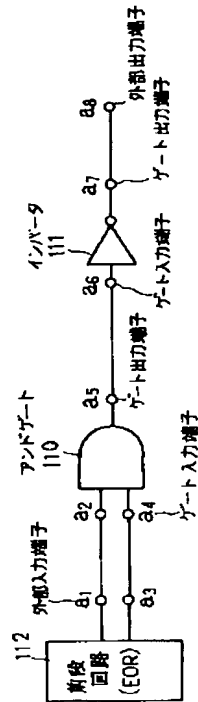
【図 16】

本発明のグラフ法による場合のアルゴリズムのフローチャート



【図 18】

従来の遅延時間算出方法の説明図



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.